IN THE CLAIMS:

Please amend the claims as follows:

1. (Original) Method to create a user profile that comprises a list of word-weight pairs or more general key structures, **characterized by** the step of computing the weights based on user features that represent a typical general behaviour of an individual user in respect to the application where the user profile is used.

2. (Original) Method according to claim 1, **characterized by** the step of evaluating an influence of the user features from a user history.

3. (Currently Amended) Method according to claim 1 ~~or 2~~, **characterized by** the step of dividing the user profile into sub user profiles according to characteristics of stereotype user profiles.

4. (Currently Amended) Method according to ~~anyone of the preceding claims~~ claim 1, **characterized by** the step of determining words to be included in the list of word-weight pairs on basis of words included in program descriptions of previous selections by the user.

5. (Original) Method according to claim 4, **characterized by** the step of determining further words to be included in the list of word-weight pairs on basis of a database evaluation of co-occurrences of words already included in the list.

6. (Currently Amended) Method according to claim 4 ~~or 5~~, **characterized by** the step of computing a weight of a word to be included in the list of word-weight pairs on basis of a comparison of an inverse document frequency of said word with respect to all available program descriptions and an inverse document frequency of said word with respect to program descriptions of previous selections by the user.

7. (Currently Amended) Method according to claim 4, ~~5 or 6,~~ **characterized by** the step of computing weights of words to be included in the list of word-weight pairs as a product of

- importance of the word with respect to all program descriptions of previous selections by the user,

and at least one of the following terms:

- a constant term, in particular one,
- importance of the word with respect to the program description in which the word is included,
- importance of the word with respect to all available program descriptions, and
- a correctional factor that depends on the length of the program description in which the word is included and the frequency of the word in this program description.

8. (Original) Method according to claim 7, **characterized in that** the importance of a word with respect to all program descriptions of previous selections by the user is determined on basis of the portion of the word in the set of all words of the program descriptions of previous selections by the user.

9. (Original) Method according to claim 8, **characterized in that** the importance of a word with respect to all program descriptions of previous selections by the user is determined under consideration of the program description in which the word is included.

10. (Currently Amended) Method according to claim 7, ~~8 or 9,~~ **characterized in that** the importance of the word with respect to the program description in which the word is included is determined on basis of occurrences of the word in said program description.

11. (Currently Amended) Method according to ~~anyone of claims 7, to 10~~ claim 7, **characterized in that** the importance of the word with respect to all available program descriptions is determined on basis of its inverse document frequency in respect to all available program descriptions.

12. (Original) Method according to claim 11, **characterized in that** the importance of a word with respect to all available program descriptions is determined under consideration of the program description in which the word is included.

13. (Currently Amended) Method according to ~~anyone of claims 7 to 12~~ claim 7, **characterized by** the step of computing weights of words to be included in the list of word-weight pairs according to the following formula:

$$w_i(t) = (1 + \frac{1}{2}\log(\frac{f_{i,t}}{f_i + 1} + 1)) \cdot a \cdot \log(\frac{N - f_t}{f_t}) \cdot (\frac{f_{i,t}}{f_{i,t} + \sqrt{f_i} / avg(\sqrt{f_i})}),$$

wherein $w_l(t)$ is the weight of a word t in a program description i, a is importance of the word t in the all program descriptions of previous selections by the user, t is the word, $f_{l,t}$ is the number of occurrences of the word t in the program description i, f_l, is the document length of the program description i, f_t, is the number of program descriptions containing the word t, and N is the number of program descriptions in the database.

14. (Currently Amended) Method according to ~~anyone of claims 7 to 13~~ claim 7, **characterized by** the step of normalizing a weight of a word included in the list of word-weight pairs by a behaviour of the user to stay with a selection and to switch selections off or to switch on/off just for this selection.

15. (Currently Amended) Method according to ~~anyone of claims 7 to 14~~ claim 7, **characterized by** the step of normalizing a weight of a word included in the list of word-weight pairs by a behaviour of the user when to actually consume a recorded selection.

16. (Currently Amended) Method according to ~~anyone of claims 7 to 15~~ claim 7, **characterized by** the step of averaging the weights $w_I(t)$ of a word over all program descriptions of previous selections by the user in which the word occurs to a weight w(t) of the new profile.

17. (Currently Amended) Method according to ~~anyone of claims 7 to 16~~ claim 7, **characterized by** the step of excluding words from the list of word-weight pairs which show a weight below a predetermined threshold or within a predetermined lower range of weights.

18. (Currently Amended) Method according to ~~anyone of claims 6 to 17~~ claim 6, **characterized by** the step of computing a weight of a word to be included in the list of word-weight pairs by considering negative selections of the user.

19. (Currently Amended) Method according to ~~anyone of claims 6 to 18~~ claim 6, **characterized by** the step of computing a weight of a word to be included in the list of word-weight pairs by considering a maximum frequency of possible user selections in respect to certain user features.

20. (Currently Amended) Method according to ~~anyone of claims 4 to 19~~ claim 4, **characterized in that** the created user profile is used as a query in possible future program descriptions to suggest at least one possible future selection to the user.

21. (Original) Method according to claim 20, **characterized in that** for each possible future program description a modified OKAPI weight is computed, wherein each matching word has its weight as a co-factor so that the search result is influenced according to the user profile.

22. (Original) Method according to claim 21, **characterized in that** said modified OKAPI weight is computed according to the following formula:

$$OKA_{\text{modified}}(q,i) = \sum_{\tau \in q \wedge \tau \in i} \omega(\tau) \cdot \log(\frac{N - f_t}{f_t}) \cdot (\frac{f_{i,t}}{f_{i,t} + \sqrt{f_i} / avg(\sqrt{f_i})}),$$

wherein w(t) is the weight of a word t over all program descriptions of previous selections by the user, q is a query built according to the user profile, i are all program descriptions of possible future selections, w(t) is the weight of the word t, $f_{I,t}$ is the number of occurrences of the word t in the program description i, f_I is the document length of the program description i, f_t is the number of program descriptions containing the word t, and N is the number of program descriptions in the database.

23. (Original) Method according to claim 21, **characterized in that** said modified OKAPI weight is computed according to the following formula:

$$OKA_{\text{modified}}(q,i) = \sum_{\tau \in q \wedge \tau \in i} \omega(\text{t}) \cdot (1 + \frac{1}{2} \log \cdot (\frac{f_{i,t}}{f_i + 1} + 1)),$$

wherein w(t) is the weight of a word t over all program descriptions of previous selections by the user, q is a query built according to the user profile, i are all program descriptions of possible future selections, w(t) is the weight of the word t, $f_{I,t}$ is the number of occurrences of the word t in the program description i, and f_I is the document length of the program description i.

24. (Currently Amended) Method according to ~~anyone of claims 20 to 23~~ claim 20, **characterized in that** the query is built by combining a stereotype user profile and the created user profile, wherein initially, when providing the suggestion for the first time, the stereotype user profile is used alone, within a first predetermined period during the collection of data for computing the created user profile a linear combination of both user profiles is used, and after said first predetermined period during the collection of data for computing the created user profile the created user profile is used.

25. (Original) Method according to claim 24, **characterized in that** the stereotype user profile comprises a generic stereotype profile which describes an average interest in everything and at least one specific stereotype profile each defining interests of a user who focuses on a specific topic.

26. (Original) Method according to claim 25, **characterized in that** a weight with which a specific stereotype user profile is considered during the linear combination to provide said suggestion is computed by matching the user selections with the data in the respective specific stereotype and increasing the weight of the stereotype if a match is found.

27. (Currently Amended) Method according to claim 24, ~~25 or 26, characterized in that~~ wherein after a second predetermined period the query is built by an individual user profile created ~~according to anyone of claims 29 to 32~~ from a multi-user profile that comprises a list of word-weight pairs, **characterized by** the step of at least once splitting the multi-user profile according to user features that represent a typical general behaviour of an individual user in respect to the application where the user profile is used.

28. (Currently Amended) Method according to ~~anyone of claims 20 to 27~~ claim 20, **characterized in that** the query is built under consideration of a filtered user profile ~~according to claim 33~~ by filtering

- a user history which is used to create the user profile, and/or
- the user profile, and/or
- the suggestion results

based on an actual situation of the user represented on basis of user features that represent a typical general behaviour of an individual user in respect to the application where the user profile is used.

29. (Original) Method to create an individual user profile from a multi-user profile that comprises a list of word-weight pairs, **characterized by** the step of at least once splitting the multi-user profile according to user features that represent a typical general behaviour of an individual user in respect to the application where the user profile is used.

30. (Original) Method according to claim 29, **characterized in that** for splitting the multi-user profile the following steps are performed:

e) performing a tentative split according to user features to generate a first and a second sub user profiles,

f) calculating the relative difference between the two sub user profiles,

g) performing steps a) and b) until all or a predetermined number of tentative splits are performed, and

h) splitting the multi-user profile according to that tentative split that yields the highest relative difference in case said relative difference lies above a predetermined threshold.

31. (Original) Method according to claim 30. **characterized in that** said relative difference is calculated by calculating a difference of a first discrete probability distribution of the first sub user profile over the user features that are contained therein and of a second discrete probability distribution of the second sub user profile over the user features that are contained therein.

32. (Original) Method according to claim 31, **characterized in that** said difference of said two discrete probability distributions is calculated using the symmetrized Kulback-Leibler-distance sum, where events which happen zero times are replaced by one virtual occurrence or where only events which happen at least once in both distributions are taken into account.

33. (Original) Method to specify a suggestion for a next selection of a user, which suggestion is determined on basis of suggestion results which are computed of future program descriptions and a user profile, **characterized by** the step of filtering

- a user history which is used to create the user profile, and/or
- the user profile, and/or
- the suggestion results

based on an actual situation of the user represented on basis of user features that represent a typical general behaviour of an individual user in respect to the application where the user profile is used.

34. (Currently Amended) Method according to ~~anyone of the preceding claims~~ claim 1 **characterized in that** said general key structure includes a forgetting factor.

35. (Currently Amended) Method according to ~~anyone of the preceding claims~~ claim 1, **characterized in that** a future program comprises a stored personal content.

36. (Currently Amended) Method according to ~~anyone of the preceding claims~~ claim 1, **characterized in that** it is used in an audio/video program suggestion engine.

37. (Original) Method according to claim 36, **characterized in that** said audio/video program suggestion engine is internet based.

38. (Currently Amended) Method according to ~~anyone of the preceding claims~~ claim 1, **characterized in that** it is client based.

39. (Currently Amended) Method according to ~~anyone of the preceding claims~~ claim 1, **characterized in that** said user features comprise one or more of the following features:

- preferred channel of audio/video program consumed by the user,
- typical time to consume an audio/video program by the user,
- length of consuming an audio/video program by the user in relation to the total length of the audio/video program,
- time of beginning the consume of an audio/video program by the user in relation to a start time of the audio/video program,
- typical length of consuming an audio video program by the user in relation to the time of consuming,
- relation between how often a particular audio/video program is consumable and how often it is consumed by the user,
- general audio/video program consuming behaviour of the user, in particular in relation to a switch-on time and length of a used audio/video device,
- audio/video programs recorded by the user,
- time duration between the recording of a particular audio/video program by the user and the consuming of said audio/video program by the user,
- actual mood of the user,
- actual wish of audio/video program entered by the user,

- year of production of an audio/video program consumed by the user,
- director and/or actor and/or group of actors of an audio/video program consumed by the user,
- type of an audio/video program consumed by the user, and
- title of an audio/video program consumed by the user.

40. (Currently Amended) Computer program product, comprising computer program means adapted to perform the method steps as defined in ~~anyone of the preceding claims~~ claim 1 when being executed on a computer, micro processor, digital signal processor, or the like including home server, set-top-box, TV, VCR, PDA.

41. (Original) Computer readable storage medium, storing thereon a computer program product according to claim 40.

42. (Currently Amended) Profiler to create a user profile that comprises a list of word-weight pairs, **characterized by** being adapted to perform the method steps as defined in ~~anyone of the preceding claims 1 to 39~~ claim 1.

43. (Currently Amended) Suggestion engine to specify a suggestion for a next selection of a user, which suggestion is determined on basis of suggestion results which are computed of future program descriptions and a user profile, **characterized by** being adapted to perform the method steps as defined in ~~anyone of the preceding claims 33 to 37~~ claim 33.

44. (Currently Amended) Suggestion engine according to claim 43, **characterized by** a profiler ~~according to claim 43~~ to perform the step of computing the weights based on user features that represent a typical general behaviour of an individual user in respect to the application where the user profile is used.